PROJECT DESCRIPTION

CRITICAL CYBERSECURITY HYGIENE

Patching the Enterprise

Murugiah Souppaya
Kevin Stine

National Institute of Standards and Technology

Mark Simos
Sean Sweeney

Microsoft

Karen Scarfone

Scarfone Cybersecurity

Final

March, 2020

cyberhygiene@nist.gov

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity challenges. Through this collaboration, the NCCoE develops modular, easily adaptable example cybersecurity solutions demonstrating how to apply standards and best practices using commercially available technology. To learn more about the NCCoE, visit https://www.nccoe.nist.gov. To learn more about NIST, visit https://www.nist.gov.

This document describes a problem that is relevant to many industry sectors. NCCoE cybersecurity experts will address this challenge through collaboration with a community of interest, including vendors of cybersecurity solutions. The resulting reference design will detail an approach that can be incorporated across multiple sectors.

ABSTRACT

Cyber hygiene describes recommended mitigations for the small number of root causes responsible for many cybersecurity incidents. Implementing a few simple practices can address these common root causes. Patching is a particularly important component of cyber hygiene, but existing tools and processes are frequently insufficient to rapidly mitigate this risk in many environments and situations. The objective of this project is to demonstrate a proposed approach for improving enterprise patching practices for general IT systems. Commercial and open source tools will be used to aid with the most challenging aspects of patching, including system characterization and prioritization, patch testing, and patch implementation tracking and verification. These tools will be accompanied by actionable, prescriptive guidance on establishing policies and processes for the entire patching life cycle, in the form of a freely available NIST Cybersecurity Practice Guide.

KEYWORDS

cyber hygiene; incidents; patching; security hygiene; software updates; vulnerabilities

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

TABLE OF CONTENTS

1 Executive Summary

Purpose

This document defines a National Cybersecurity Center of Excellence (NCCoE) project focused on helping organizations rapidly and effectively improve their security hygiene. The project's objective is to increase cybersecurity ecosystem resiliency by helping organizations to overcome the resource-intensive and often thankless nature of security hygiene. The project aims to increase awareness of the importance of security hygiene issues, recommend specific prioritized actions to overcome common obstacles, and establish a natural glide path for organizations to continue on to achieve a comprehensive security hygiene program based on existing standards, guidance, and publications.

The driver behind security hygiene is that there are a relatively small number of root causes for many data breaches, malware infections, and other security incidents. Implementing a few relatively simple practices can address those root causes to prevent many incidents from occurring and to lower the potential impact of incidents that still occur. In other words, security hygiene practices make it harder for attackers to succeed and reduce the damage they can cause.

Unfortunately, security hygiene is easier said than done. For example, information technology (IT) professionals have known for decades that patching software—operating systems, applications, and the like—eliminates known vulnerabilities. Even though there is widespread recognition that patching can be incredibly effective at mitigating security risk, patching is often resource-intensive, and the act of patching itself can reduce system and service availability. Attempts to reduce resource utilization and expedite patch distribution, such as not testing patches before production deployment, can inadvertently break system functionality and disrupt operations, in some cases causing a significant negative impact to the organization. On the other hand, delays in patch deployment create a larger window of opportunity for attackers.

Patching is a particularly important component of cyber hygiene, but existing tools are insufficient for many environments and situations. For example, many organizations lack tools to help them measure and assess the effectiveness and timeliness of their patching efforts. Many organizations also struggle to prioritize patching efforts, test patches before deployment, and adhere to policies for how quickly patches need to be applied in different situations.

How, when, and what to patch can be difficult decisions for any organization. Each organization must balance security with mission impact and business objectives, and figure out their risk tolerance for each. Recent cybersecurity attacks have highlighted the dangers of having equipment that has not been patched. Even with recent events and the historical attacks that have been successfully carried out due to unpatched systems, patching remains a problem.

This project will result in a publicly available NIST Cybersecurity Practice Guide, a detailed implementation guide of the practical steps needed to implement a cybersecurity reference design that addresses this challenge.

Scope

The objective of this building block project is to demonstrate a proposed approach for improving enterprise patching practices for general IT systems. In this project, commercial and open source tools will be used to aid with the most challenging aspects of patching, including system characterization and prioritization, patch testing, and patch implementation tracking and

verification. These tools will be accompanied by actionable, prescriptive guidance on establishing policies and processes for the entire patching life cycle, to include defining roles and responsibilities for all affected personnel, and establishing a playbook with rapid mitigation actions for destructive malware outbreaks that organizations can execute tactically in the first 30 days, and recommendations that can be implemented strategically beyond 30 days.

The scope of this building block is general IT systems. There are additional challenges with patching for legacy IT systems and virtual systems, as well as industrial control systems (ICS), Internet of Things (IoT) devices, and other technologies stemming from operational technology (OT). Future work could add some or all of these system types to the building block.

All aspects of security hygiene other than those related to patching are out of the scope of this building block. The NCCoE is considering adding other security hygiene elements to this building block in the future. Examples include disabling unneeded legacy protocols, only using current (supported) versions of operating systems and applications, and protecting privileged access.

Assumptions/Challenges

The primary technical elements of this project are as follows:

- IT endpoints (desktops/laptops and servers running commonly used modern operating systems and applications, including virtual machines and containers)
- Mobile devices
- Networking devices (such as routers and switches)
- Network firewalls
- Patch management systems
- Intrusion detection and prevention systems

An IT endpoint for an enterprise would have firmware, operating system(s), and application(s) to be patched. The endpoint may be in a fixed location within the organization's own facilities or in a fixed location at a third-party facility (e.g., a data center), or it may be intended for use in multiple locations, such as a laptop used at the office and for telework. The proposed approach for improving enterprise patching practices would have to account for all of these possibilities.

Problems sometimes occur with patches, such as a failure during installation, a patch that cannot take effect until the endpoint is rebooted, or a patch that is uninstalled because of operational concerns or because an attacker wants to maintain a vulnerability in a compromised system. This project follows a "trust but verify" philosophy that does not assume installing a patch automatically means the patch is successfully and permanently applied.

There are no standard protocols, formats, etc. for patch management, including patch distribution, integrity verification, installation, and installation verification. It is also highly unlikely for a single patch management system to be able to handle all patch management responsibilities for all software on IT endpoints. For example, some applications may handle patching themselves and not be capable of integrating with a patch management system for patch acquisition and installation.

Background

Patching is not a new challenge for organizations. Many patching guidelines have been published over the years. NIST released Special Publication (SP) 800-40, *Procedures for Handling Security Patches*, in 2002 [1]. Since then, two revisions of SP 800-40 have been published. SP

800-40 Version 2, *Creating a Patch and Vulnerability Management Program*, includes discussion of creating and managing such a program, and testing its effectiveness [2]. The latest revision, SP 800-40 Revision 3, *Guide to Enterprise Patch Management Technologies*, was released in 2013 [3]. It is focused on assisting organizations in understanding the basics of enterprise patch management technologies and increasing the automation of mature patch management programs.

Another noteworthy publication is SP 800-184, *Guide for Cybersecurity Event Recovery*, which provides recommendations for rapid recovery from incidents when they occur and helps to minimize the impact on the organization and its constituents [4]. NIST SPs 800-40v2, 800-40r3, and 800-184 can be leveraged to develop a playbook around patching as a recovery step in the event of a fast destructive malware outbreak like Petya or WannaCrypt.

In addition to having practices in place for patching, organizations also need inventory capabilities so that at any time, the organization knows what IT systems it has, what dependencies each system has on other systems, what the criticality of each system is, and what the impact would be of a system compromise or operational failure. Without this information, patching efforts may be significantly hampered. Gathering and maintaining this information in a timely manner necessitates relying on tools.

2 SCENARIOS

Scenario 0: Asset identification and assessment

This scenario identifies the assets and classifies them based on different impact levels to prioritize the order of remediation. It leverages free and commercial tools that can be used to discover assets across the enterprise and the cloud to enumerate firmware, operating systems (OSs), and applications.

Knowing which software and software versions are in use and predetermining remediation priorities are critically important to all other patching processes. Without accurate, up-to-date, and comprehensive information, an organization will have difficulties effectively and efficiently performing patching processes, thus increasing risk. While many enterprises have constant asset attrition, it is important to have full and accurate inventory of critical assets and the best possible inventory for the full enterprise.

Scenario 1: Routine patching

This is the standard procedure for patches that are on a regular release cycle and haven't been elevated to an active emergency status (because of active exploit in the wild or extreme vulnerability severity). This includes endpoint firmware, OS, and applications, server OS and applications hosted on-premises or in the cloud (e.g., Infrastructure as a Service), as well as "network devices" like firewalls, Storage Area Network (SAN) devices, routers, network switches, and other network appliances.

Most patching falls under this scenario or Scenario 2. However, because routine patching does not have the urgency of emergency patching, and routine patch installation can interrupt operations (e.g., device reboots), it is often postponed and otherwise neglected. This provides many additional windows of opportunity for attackers.

Scenario 2: Routine patching with cloud delivery model

This is the standard procedure for patches that are delivered through a cloud delivery model, such as a mobile device or a "Windows as a Service (WaaS)" model with Windows operating systems, Apple Software Update, and mobile device software updates for Android and iOS devices provided by device manufacturers or mobile operators.

This scenario is similar in importance to Scenario 1, Routine Patching. However, organizations may not be as accustomed to cloud-delivered patches (which are frequently cumulative for the whole system vs. discrete patches), so this scenario is somewhat more likely to be overlooked by organizations, which increases risk.

Scenario 3: Emergency patching

This is the emergency procedure to address active patching emergencies in a crisis situation, such as extreme severity vulnerabilities like MS17-010, as well as vulnerabilities that are being actively exploited in the wild. The scope of targets is the same as scenario 1.

Emergency patching needs to be handled as efficiently as possible to prevent imminent exploitation of vulnerable devices. Key characteristics include identifying vulnerable assets, triaging and applying patches based on a priority list, and tracking and monitoring the state of those assets.

Scenario 4: Emergency workaround (and backout if needed)

This is the emergency procedure in a crisis situation to temporarily mitigate risk for vulnerabilities prior to a vendor releasing a patch. It is typically required when the vulnerability is being actively exploited in the wild. The workaround can vary and may or may not need to be rolled back afterward. The scope of targets is the same as scenario 1.

Organizations need to be prepared to quickly implement a wide variety of emergency workarounds to protect vulnerable devices. Without processes, procedures, and tools in place to implement workarounds, too much time may be lost and vulnerable devices may be compromised before workarounds are in place. This may require disabling system functionality, having automated mechanisms to apply these changes, and having capabilities to revert back these changes when a permanent and approved patch is released.

Scenario 5: Isolation of unpatchable assets

This is the reference architecture and implementation of isolation methods to mitigate the risk of systems which cannot be easily patched. This is typically required if routine patching is not able to accommodate these systems within a reasonable timeframe (usually X days or less). Most systems in this scope are legacy unsupported systems or systems with very high operational uptime requirements.

Isolation is a form of workaround that can be highly effective at stopping threats against vulnerable devices. Organizations need to be prepared to implement isolation methods when needed and to undo the isolation at the appropriate time to restore regular device access and functionality.

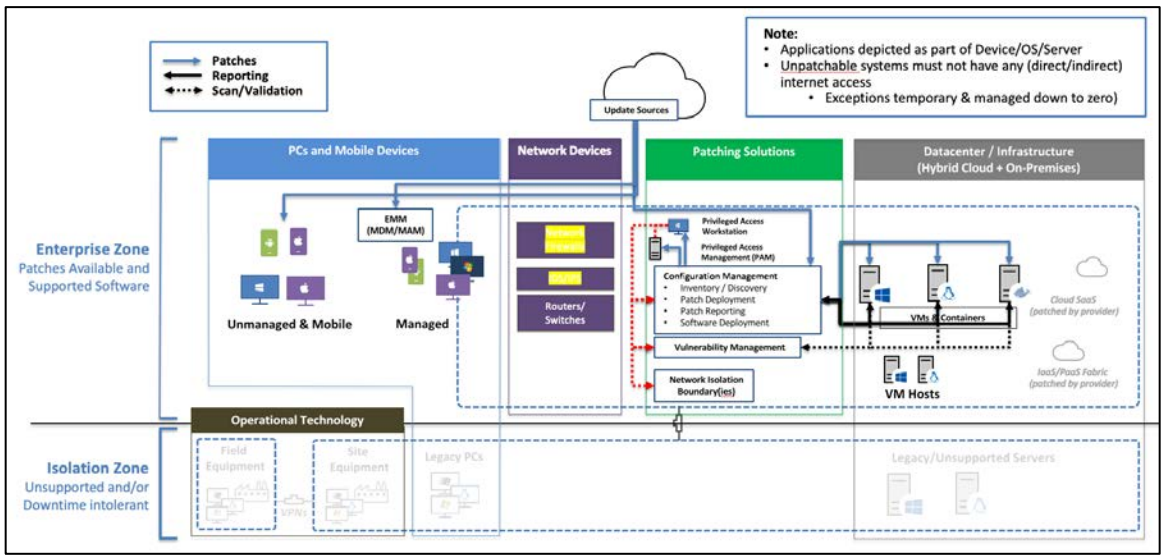
Scenario 6: Patch management system security (or other system with administrative privileges)

This is a reference architecture and implementation of recommended security practices for systems like patch management systems which have administrative privileges over many

systems. This will include practices like least privilege, privileged access workstations, and software updates.

3 HIGH-LEVEL ARCHITECTURE

Figure 1: Security Patching Reference Architecture



Patching is a relatively simple operation of updating existing software, but the implementation of the systems has a small amount of complexity. Core assumptions of the architecture depicted in Figure 1 include:

- Unpatchable systems must not have any internet access (direct/indirect).
- Any exceptions are temporary and quickly managed down to zero.

You must patch all the software on the network, including operating systems across devices and servers, applications across devices and servers, and firmware in the devices/hardware. It is critical not to overlook that network, storage, and other enterprise devices also run operating systems and firmware and must be patched regularly. Figure 1 depicts the common enterprise components that need to be regularly updated and maintained.

The critical cyber hygiene initiative is focused first on common enterprise services in the IT environment. Operational technology and IoT devices are out of scope for the first phase not because of lack of importance, but to ensure rapid delivery of the most common components.

The patching system components include:

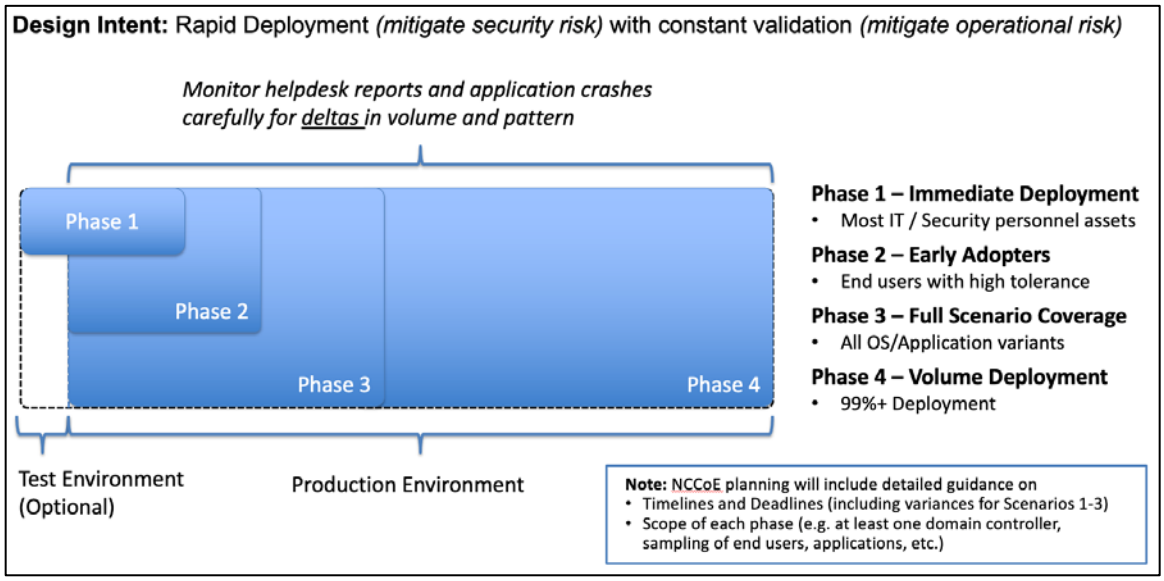
- Configuration management tools (where patching is usually managed, though sometimes standalone services like Windows Server Update Services [WSUS] are also available)
- Vulnerability assessment to provide independent assessment of whether updates are applied correctly (plus detect other non-update vulnerabilities)
- Security components for the patching and configuration management infrastructure (elevated security required, given the potential enterprise-wide impact of compromise)

- Network isolation boundaries that protect systems from attacks on eternally unpatched vulnerabilities (unsupported, sensitive to operational downtime, etc.)

Note that the patching by a cloud provider is a "trust but verify" situation where the cloud provider has to take care of the day-to-day responsibility, but you as a customer should have the ability to check on this. The mechanisms for how to do this can vary (during acquisition, informal/formal processes, etc.,) but many compliance regimes require service providers to provide access to audit reports.

The reference security patching process shown in Figure 2 allows you to maximize deployment speed while limiting the risk of application incompatibility. Note that measuring patch impact should focus on the changes to volume and pattern of likely issues (helpdesk calls and application crash/error reports). This process should be consistent regardless of the speed of the deployment (measured in the ideal of hours/days or starting out measuring in weeks).

Figure 2: Security Patching Process



The following describe each of the phases depicted in Figure 2.

Phase 1: **Immediate Deployment**

The goal of this is to immediately test the updates against real-world scenarios with technologically savvy users (who are also stakeholders in patching) in the IT and security organizations.

Target**:** Most IT/security personnel assets

Phase 2: **Early Adopters**

The goal of this is to rapidly include as many scenarios and technical profiles to flush out application compatibility issues. To mitigate the potential of operational downtime or interruption, we recommend recruiting early adopter users across the business with a high tolerance for interruption (and possibly including 'dummy' versions of production systems like process control network PCs, etc.) While it is desirable to cover all OS/application profiles, it is acceptable not to do so in this stage.

Targets**:** End users with high tolerance; "dummy" systems with production applications installed but no operational dependency

Phase 3: **Full Scenario Coverage**

The goal of this is to cover all OS/application profiles to create high confidence for enterprise rollout. This group may need to evolve as business needs and application profiles change, so including the update of this group in change release processes is highly recommended.

Target**:** All OS/application profiles

Phase 4: **Volume Deployment**

The goal of this phase is to achieve as close to 100% coverage of the update as feasible so the organization's security attack surface does not include known vulnerabilities that an attacker could exploit at extremely low cost to them.

Target**:** 99%+ deployment

Component List

The high-level architecture will include the following components:

- **PCs and Mobile Devices** – The architecture will include the following components used on the client side:
    - **Managed**: There will be numerous enterprise PCs (desktops and laptops) in use that are managed by the organization and need their operating systems patched.
    - **Unmanaged & Mobile:** There will be numerous unmanaged PCs (desktops and laptops) and mobile devices in use within the organization that need their operating systems patched.
    - **Apps:** The apps on the managed PCs, unmanaged PCs, and mobile devices will need to be patched or updated.
    - **PC Firmware:** The firmware on the managed and unmanaged PCs will need to be patched or updated.
    - **EMM (MDM/MAM):** There will be an Enterprise Mobility Management (EMM) solution deployed to help manage the mobile devices, including identifying vulnerabilities and applying patches and updates. The EMM will be paired with Mobile Device Management (MDM) and Mobile Application Management (MAM) solutions for the mobile device platforms in use.
- **Network Devices** – The architecture will include the following components providing network-based services for other parts of the architecture:
    - **Firewalls:** Firewalls will restrict network traffic between networks and network segments.
    - **IDS/IPS:** Intrusion detection systems (IDS) and intrusion prevention systems (IPS) will monitor network traffic for malicious packets and behaviors, and may block or alert on the traffic.
    - **Routers/Switches:** Routers and switches will help direct network traffic from source to destination and may impose some basic restrictions on the traffic.

    - **Storage:** Network-based storage systems will provide data storage for other components on the architecture.
- **Update Sources** – Components of the architecture will interact with external update sources controlled and managed by third parties.
- **Patching Solutions** – The architecture will include the following components used to perform patching responsibilities:
    - **Privileged Access Management (PAM) System:** The PAM system will be used to help manage and monitor privileged access to other systems, most notably the configuration management and vulnerability management systems.
    - **Privileged Access Workstation:** The privileged access workstation is a PC (desktop or laptop) that will be authorized to administrate the configuration and vulnerability management systems via the PAM system.
    - **Configuration Management System:** The configuration management system will be used for several purposes, including inventory/discovery, patch deployment, patch reporting, and software deployment.
    - **Vulnerability Management System:** The vulnerability management systems scan for software vulnerabilities and assist with managing these.
    - **Network Isolation Boundaries:** The network controls isolate systems to mitigate the risk of exploitation from another networked system.
- **Datacenter/Infrastructure (Hybrid of Cloud + On-Premises)** – The architecture will include the following components used to provide servers and server infrastructure:
    - **Apps:** There will be numerous applications running on both cloud and on-premises servers, and these applications will need to be patched.
    - **VMs & Containers:** There will be virtual machines (VMs) and container technologies running on both cloud and on-premises VM hosts. The VMs and container technologies will need to be patched.
    - **VM Hosts:** There will be numerous VM hosts, which are the physical servers the VMs and containers run on top of. The hosts will need their firmware patched.
    - **Server/Other Firmware:** The VM hosts and other physical servers (e.g., on-premises) will need their firmware patched or updated.
    - **Cloud Software as a Service (SaaS) and Infrastructure as a Service (IaaS)/Platform as a Service (PaaS) Fabric:** The resources provided by cloud providers will be patched by the providers.

A more detailed architecture and design will be developed once the project is approved and the project team has been assembled.

Desired Requirements

An NCCoE build for this project will require the following components:

- PCs and mobile devices, including operating systems, firmware, and apps
- EMM, MDM, and MAM solutions
- Firewalls and intrusion detection and prevention systems
- Routers/switches
- Network-based storage

- Update sources
- PAM system and privileged access workstation
- Configuration management system
- Vulnerability management system
- On-premises datacenter and cloud infrastructure, including servers, VM hosts, VMs, containers, apps, and firmware

4 Relevant Standards and Guidance

The resources and references required to develop this solution are generally stable, well understood, and available in the commercial off-the-shelf market.

Secure Update Guidelines

- NIST Special Publication (SP) 800-40 Version 2, *Creating a Patch and Vulnerability Management Program*. See https://doi.org/10.6028/NIST.SP.800-40ver2
- NIST Special Publication (SP) 800-40 Revision 3, *Guide to Enterprise Patch Management Technologies*. See https://doi.org/10.6028/NIST.SP.800-40r3
- NIST Special Publication (SP) 800-184, *Guide for Cybersecurity Event Recovery*. See https://doi.org/10.6028/NIST.SP.800-184
- Department of Homeland Security (DHS), Binding Operational Directive 19-02, *Vulnerability Remediation Requirements for Internet-Accessible Systems*. See https://cyber.dhs.gov/bod/19-02/
- DHS, Binding Operational Directive 16-02, *Threat to Network Infrastructure Devices*. See https://cyber.dhs.gov/bod/16-02/

Microsoft Software Update Guides

- Microsoft, *Security Update Guide.* See https://portal.msrc.microsoft.com/en-us/
- Microsoft, *Microsoft Lifecycle Policy*. See https://support.microsoft.com/en-us/lifecycle
- Microsoft, *Quick Guide to Windows as a Service*. See https://docs.microsoft.com/en-us/windows/deployment/update/waas-quick-start

5 Security Control Map

This table maps the characteristics of the commercial products that the NCCoE will apply to this cybersecurity challenge to the applicable standards and best practices described in the Framework for Improving Critical Infrastructure Cybersecurity (Cybersecurity Framework) [5], and other NIST activities. This exercise is meant to demonstrate the real-world applicability of standards and best practices, but does not imply that products with these characteristics will meet your industry's requirements for regulatory approval or accreditation.

Table 1: Security Control Map

Cybersecurity Framework Category	Cybersecurity Framework Subcategory	Draft SP 800-53 Revision 5 Controls [6]
Asset Management (ID.AM): The data, personnel, devices, systems, and facilities that enable the organization to achieve business purposes are identified and managed consistent with their relative importance to organizational objectives and the organization's risk strategy.	**ID.AM-1:** Physical devices and systems within the organization are inventoried	CM-8, System Component Inventory
	ID.AM-2: Software platforms and applications within the organization are inventoried	CM-8, System Component Inventory
	ID.AM-4: External information systems are catalogued	SA-9, External System Services
Risk Assessment (ID.RA): The organization understands the cybersecurity risk to organizational operations (including mission, functions, image, or reputation), organizational assets, and individuals.	**ID.RA-1:** Asset vulnerabilities are identified and documented	CA-7, Continuous Monitoring RA-3, Risk Assessment RA-5, Vulnerability Scanning SI-2, Flaw Remediation
Data Security (PR.DS): Information and records (data) are managed consistent with the organization's risk strategy to protect the confidentiality, integrity, and availability of information.	**PR.DS-2:** Data-in-transit is protected	SC-8, Transmission Confidentiality and Integrity
	PR.DS-6: Integrity checking mechanisms are used to verify software, firmware, and information integrity	SI-7, Software, Firmware, and Information Integrity
Information Protection Processes and Procedures (PR.IP): Security policies (that address purpose, scope, roles, responsibilities, management commitment, and coordination among organizational entities), processes, and procedures are maintained and used to manage protection of information systems and assets.	**PR.IP-12:** A vulnerability management plan is developed and implemented	RA-3, Risk Assessment RA-5, Vulnerability Scanning SI-2, Flaw Remediation

APPENDIX A REFERENCES

[1] NIST, Special Publication (SP) 800-40, Procedures for Handling Security Patches, 2002.

[2] NIST, Special Publication (SP) 800-40 Version 2, Creating a Patch and Vulnerability Management Program, 2005. https://doi.org/10.6028/NIST.SP.800-40ver2

[3] NIST, Special Publication (SP) 800-40 Revision 3, Guide to Enterprise Patch Management Technologies, 2013. https://doi.org/10.6028/NIST.SP.800-40r3

[4] NIST, Special Publication (SP) 800-184, Guide for Cybersecurity Event Recovery, 2016. https://doi.org/10.6028/NIST.SP.800-184

[5] NIST, Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1, 2018. https://doi.org/10.6028/NIST.CSWP.04162018

[6] Joint Task Force, NIST Draft Special Publication (SP) 800-53 Revision 5, Security and Privacy Controls for Information Systems and Organizations, 2017. https://csrc.nist.gov/CSRC/media//Publications/sp/800-53/rev-5/draft/documents/sp800-53r5-draft.pdf

APPENDIX B ACRONYMS AND ABBREVIATIONS

DHS	Department of Homeland Security
EMM	Enterprise Mobility Management
IaaS	Infrastructure as a Service
ICS	Industrial Control System
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Prevention System
IT	Information Technology
MAM	Mobile Application Management
MDM	Mobile Device Management
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
OS	Operating System
OT	Operational Technology
PaaS	Platform as a Service
PAM	Privileged Access Management
PC	Personal Computer
SaaS	Software as a Service
SAN	Storage Area Network
SP	Special Publication
VM	Virtual Machine
WaaS	Windows as a Service
WSUS	Windows Server Update Services